An Ontology for Requirements Analysis of Managers' Policies in Financial Institutions

Feltus, Christophe; Rifaut, André

Published in:
Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007), Funchal, Portugal

DOI:
10.1007/978-1-84628-858-6_3

Publication date:
2007

Document Version
Early version, also known as pre-print

An Ontology for Requirements Analysis of Managers' Policies in Financial Institutions

C. Feltus[1] and A. Rifaut[1]
[1] Centre de Recherche Public Henri Tudor, 29, Avenue John F.Kennedy, L-1855 Luxembourg-Kirchberg, Luxembourg (http://www.tudor.lu)
{Christophe.Feltus, Andre.Rifaut}@tudor.lu

Abstract. Policies are an important organizational tool giving an effective support for building business systems, from the strategic level down to the operational and technical levels. In particular, policies are a cornerstone for the governance system of financial institutions. In international organizations, a lot of policies span all country-local representatives and span all organizational levels. This work is part of a series concerning the improvement of requirements engineering methods for process-based organizations. This requires enhancing a shared vision between employees of the process responsibilities, by advocating cross-functional thinking with the focus set to the outcomes of the processes, and defining the outcomes in relationship with the business goals. We complement the works on business process models by the managers' concerns, i.e. the managers' responsibilities for value to be delivered by the processes. This research proposes a method for constructing policy models. Ontology is defined for interoperability purposes of the models of different organizational levels. The main formal analyse that is used for verification purposes is the reliability of the policy system and its impact on the reliability of the operational system which is one important objective of recent governance regulations.

1 Introduction

Policies are an important organizational tool giving an effective support for building business systems, from the strategic level down to the operational and technical levels. In particular, policies are a cornerstone for the governance system of financial institutions. In international organizations, a lot of policies span all country-local representatives (e.g. policies addressing the organization strategy, or the international regulations) and span all organizational levels.

This work is part of a series concerning the improvement of requirements engineering methods for *process-based organizations*, in particular for financial institutions. This requires enhancing a shared vision between employees of the process responsibilities, by enhancing cross-functional thinking with the focus set to the outcomes of the processes, and defining the outcomes in relationship with the business goals [1]. We complement the works on business process models [2] by the *managers' concerns*, i.e. the managers' responsibilities for value to be delivered by the processes. This research proposes a method for *constructing those policy models*, i.e. defining sets of assigned responsibilities in the organization. Ontology is defined for interoperability purposes between the different models and grounded in a standard first-order linear temporal logic semantics when more expressive power is needed than descriptive logics [3]. The main formal analyse that can be used for verification purposes is *the reliability of the policy system and its impact on the reliability of the operational system.* Actually, reliability is one important objective of recent governance regulations [4][5].

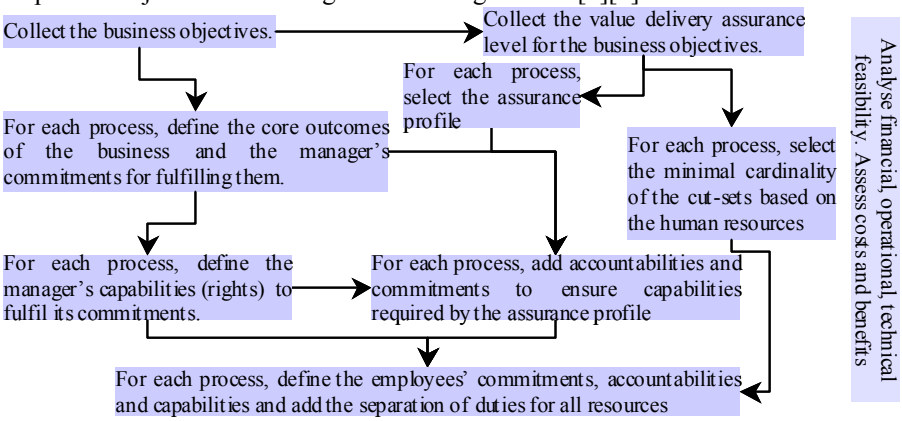

Fig. 1. Method for defining policies ensuring a value delivery assurance level

The main focus of research about policy in IT systems concerns the design of policies and the design of IT systems that efficiently operate those policies, the ability of those policies to express concepts such as *segregation of duties*, *delegation* (of rights, permissions, and obligations), *accountability*, ... [6][7]. Our proposal complements those results by easing the elicitation of requirements for managers' policies and relies on 4 principles. First, the design of policies must be done in tight relationship with objectives, strategies, and key indicators. Second, all organizational levels must be addressed including aspects outside the scope of IT systems and Information System (IS). Third, responsibilities must be fully decomposed into the capabilities (i.e. permissions and rights), the commitments (i.e. the obligations or goals to fulfil) and the accountability requirements. Fourth, policies must always be related to the enforcement of an optimal resource usage in regard to the defined objectives, strategies, and indicators. This paper details the method presented in Figure 1.

Table 1. Purpose and outcomes of the Operational Risk Mitigation/Control process

	Operational Risk Mitigation/Control (BORO.1)
Purpose	*The purpose of the Operational Risk Mitigation/Control process is to mitigate the assessed operational risks and to manage operational risk impact.*
Outcome 1	An operational risk mitigation and control strategy is developed, including the principles of how operational risk is to be mitigated and how its realization is to be control, according to the size, the sophistication, the nature and the complexity of the bank's activity;
Outcome 2	The existing option to mitigate risk are analyzed and, for each risk, the most in accordance with bank's strategy is chosen;
…	

The rest of this Section presents the case study that will be used and the aims and context of our research for process-based organizations. The ontology of operational assurance level of business process is explained in Section 2. For all step of the construction process analyses of policies are described in Section 3. Then the ontology of more IT-oriented policies is presented in Section 4, with an example how to derive those policies from managers' policies. The last section concludes on the originality of our work and presents the future works.

Our case study is based on the Basel II Accord [5] that defines the requirements of operational risk management systems that must be implemented in Banks. Those requirements have been structured in our previous works [9] (the result is freely accessible on the website of CSSF, the Luxemburg Bank Regulators, [10]; see the example in Table 1 about the operational risk mitigation process).

Our case study focuses concerns an operational risk management system implementation. The corporate operational risk management team (***CORMT***) has specialized sub-teams for each Business Unit, in particular for Venture Capital Management (i.e. ***CORMT-AM*** related to the Business Line "Asset Management") and for Securities Management (i.e. ***CORMT-CF*** related to the Business Line "Corporate Finance"). The Basel II corporate policy imposes that each business line is responsible for the day-to-day management of its own risks; however, the implementation of the risk management system is the responsibility of the corporate operational risk management team. Each business unit manager assigns clerks to collect the operational risk data.

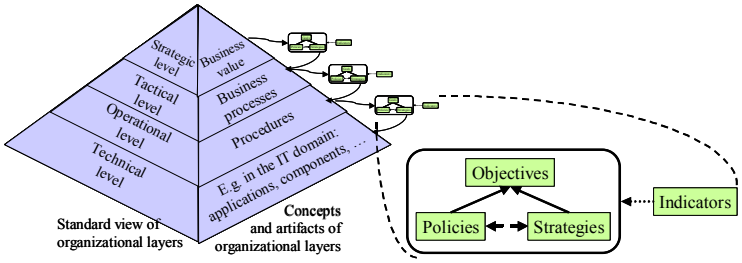

Fig. 2. The alignment of policies in management methods

1.1 Process Models for Business Process Managers

Our global approach, presented in [11], is based on the organizational pyramid [12] used in international financial institutions in order to align policies with business processes described with goal-oriented models [13]. In the context of financial regulations, a good governance system is based on the 4 organizational layers [14] (strategic, tactical, operational and technical levels on Figure 2) for aligning business value[15], business processes, procedures and technical artefacts [16] (such as IT applications in the IT domain). The core of the ontology used for this alignment is presented at the bottom of Figure 1: objectives, strategies, policies and indicators [17]. The formal definition of those 4 core concepts of the ontology relies on goal-oriented models [18][19], with their semantics presented in the context of the requirements engineering language ***i**** [20][13].

This framework allows abstracting requirements from implementation details as much as possible in order for managers being able to assess the effectiveness and efficiency of the processes in relationship with the business goals. A number of methods exist for assessing this alignment that have been created for managers, such as e.g. the Balanced Score Card [21]. Our proposal complements those methods by using a goal-oriented description of business processes similar to the one used for process models used in the standard ISO/IEC 15504 [13]. For each process, this standard imposes to define the main goal of the process (the purpose), the sub-goals (outcomes) for which an objective judgment can be made upon their fulfilment on the basis of the indicators of the outcomes. Indicators are categorized into base practices, work-products, and resources.

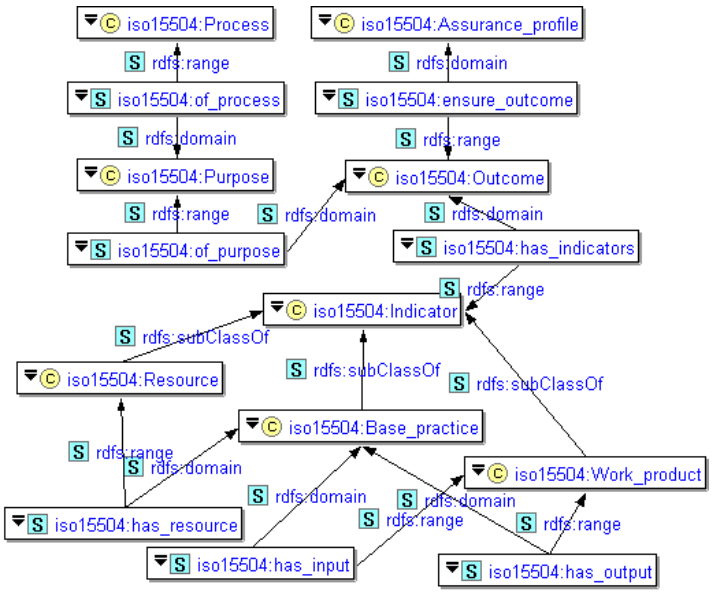

Fig. 3. Ontology of business process goal, indicators and assurance profile.

1.2 Operational assurance profile

A number of case studies [22][9][23] have shown that with this goal-oriented model, the goals concerning managers can easily be described (together with the expected outcomes) and linked to the business goals and the business system implementation goals. Indeed, in addition to process-specific outcomes shown in the preceding table, other generic outcomes (not shown) concern the process performance management (i.e. planning and monitoring, availability of resources and information, …), the work-product management (work-product control and review, …), the process definition, the process deployment, … In Section 2, those typical outcomes are the main concern of operational policies imposed by business process managers. In [9] it is shown how the interaction between processes can be precisely analyzed at that high-level of abstraction suited for business unit managers, allowing them to delegate process implementation and still being able to assess that the implemented system is faithful to the requirements. On Figure 3 is shown the ontology for modelling process goals and indicators. For clarity purposes, the concepts "process system implementation" and "goal" have not been displayed on the diagram, but the concepts "purpose", "outcome", "indicator" (refining the outcomes) and "business goal" (this latter not shown) are sub-classes of the concept "goal".

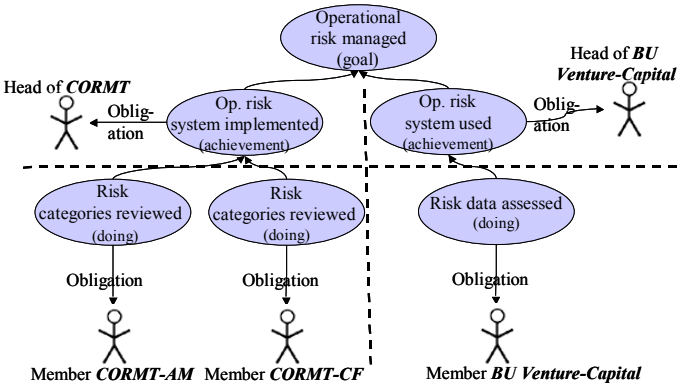

Fig. 4. Vertical responsibility refinement

With this ontology assurance profiles are sets of outcomes that defines the operational assurance of each system implementation. The generic outcomes have a comprehensive set of pre-defined indicators. Those generic outcomes and indicators are typical concerns of managers. So, it easy for them to select the best assurance level and assess its financial, organizational and technical feasibility. For instance, the "venture capital work-product management" assurance profile includes all outcomes imposing that the venture capital documents are controlled and reviewed. Process system implementations (e.g. one for each country) must be compliant to this profile. This assurance profile depends on the business domain of the business process, showing the specificities of each country-local system

implementation made by managers. When an assurance profile is selected, it is easier to constructively build an integrated set of policies (i.e. set of assigned responsibilities) ensuring the profile.

2 Policies for Business Process Managers

This section details the ontology used for defining policies that is adequate for business process managers which aim is to address the completeness of the responsibilities included in a policy to ensure the operational reliability as required by the selected operational assurance profile. Recall that a policy is a set of responsibilities, including the associated goals, and assignments to resources.

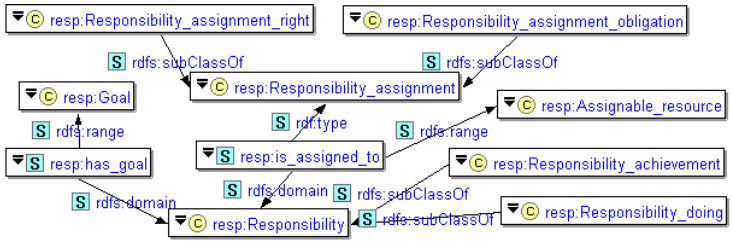

Fig. 5. Ontology of vertical responsibility refinement

2.1 Vertical responsibility decomposition

The ontology distinguishes between, first, high-level policies concerning the work of managers aiming at the creation, maintenance, and optimization of their process ("responsibility-for-achievement"), and, second, the policies imposed on, for instance, clerks when doing the work (or executing the procedures) described in the business processes ("responsibility-for-doing").

In the example shown on Figure 4, the corporate goal of managing operational risks is split into two main parts: first the implementation of the operational risk management system is assigned to the head of the ***CORMT***, and the second part, the operational use of the system is assigned to each business unit manager (e.g. the head of venture capital business unit). They have to organize the work of employees belonging to their unit: the specialized team members (***CORMT-AM*** and ***CORMT-CF***) must have the knowledge (through coaching or attendance to courses) for dealing with operational risk categories. In Figure 4, (neutral) graphical notations are used (instead of showing ontology instances) to ease the readability: ovals are responsibilities represented by their goals, arrows with labels are assignments (that will be explained hereafter), and arrows without labels are goal refinements. Recall that this is a goal-oriented model and this is why responsibilities are identified with their goal (Figure 5) just like processes are modelled with their goals decomposition (Figure 3). The three kinds of refinements in Figure 5 are just refinements of the responsibilities' goals. This is an advantage for steering processes because the goal-refinements are driven by the assurance profile and by the responsibility decomposition.

In most of requirements engineering models of business processes, the scope of the model tend to show that for the risk data being collected, it is sufficient for the clerk to (timely and accurately) input the data (and for ***CORMT-AM*** or ***CORMT-***

CF to validate the data entered). However, this is only the description of "responsibility-of-doing", i.e. the low-level operational aspect. However, depending on the level of reliability that is required, the goal is reached if and only if all "sub-responsibilities-of-doing" are fulfilled together with the "responsibilities-of-achievement". In real case studies the bottom of the tree, i.e. the "responsibility-of-doing" are rarely sufficient. Indeed, the "principle of exception" of management theory, shows that it is nearly impossible to detail all managers' responsibilities into "responsibilities-for-doing" due to unforeseen events. In the ORDIT methodology [24][25] similar concepts are presented, but not fully formalized in goals-oriented models and business process models for managers.

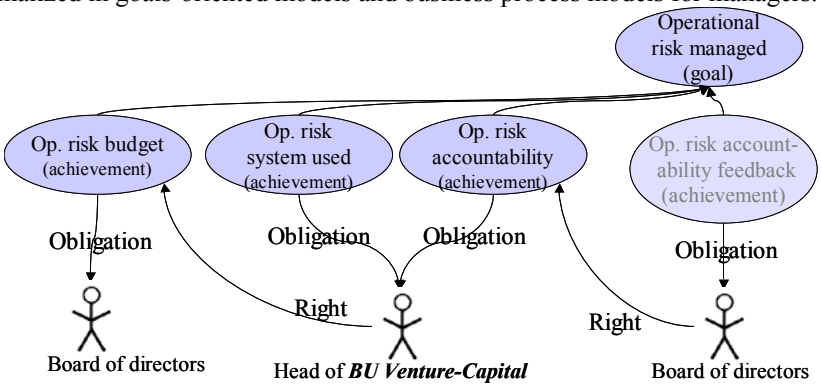

Fig. 6. Full cover of responsibility decomposition

2.2 Full cover of responsibility decomposition

When decomposing responsibilities, all 3 aspects must be covered: in order for a person being *committed* to fulfilling a responsibility and accepting to be *accountable* for that responsibility, that person always must get the right for having the *capabilities* needed to fulfil both its commitment and accountability. A missing aspect, induced weaknesses in the responsibility system decrease its reliability.

In the running example (see Figure 6), the corporate goal of having an operational risk management system can be split into the following responsibilities: the business unit manager is responsible for the day-to-day operational risk management, whereas the corporate risk manager is responsible for creating an operational risk system. Of course, both managers must have the capabilities (e.g. budgets) corresponding to these responsibilities. Moreover, in order to ensure their commitment, both managers must be accountable for their responsibilities (i.e. their commitments) and the usage of their rights (i.e. the usage of the capabilities received). Similarly, when managers are assigning "responsibilities-for-doing", then they have to provide all resources needed for employees belonging to their unit to have sufficient time allocated for operational risk data assessment.

The capabilities required are the resources needed such as some budget and manpower, the description of the tasks, input/output information, … Those capabilities are rights for the person responsible of the commitment, but it is an

obligation for another person (to provide the capabilities for the first one, see Figure 6). Accountability is important: one is accountable for both its commitment and the usage of its rights (i.e. the time usage, the access to some information, the budget, …) The 3 concepts involved in this decomposition (classes "Responsibility capability", "Responsibility commitment", "Responsibility accountability") are subclass of the class "Responsibility" (not shown on Figure 5).

3 Analyses of policy requirements models

Relying on goal-oriented models within which all formal definitions are related to goals, formal analyses are made through the analysis of the goal model. In practice, lightweight formal analysis tools such as model checkers, or automatic theorem provers are used. It has been shown that those tools, although having a limited formal analysis capability, are efficient with goal-oriented models because most of errors can be pointed out by making local analyses of the goal model. [26][27]. Our analysis tool is based on the SWI-Prolog-XPCE Semantic Web Library package [28] in conjunction of the Otter automated theorem prover [37][36] and using bounded-model checking techniques for dealing with temporal logic formulae [29]. As said above, there is no intent to have a formal proof of the properties (hence, no intent to prove the inexistence of counter-examples), but the intent is to exhibit counter-examples. Formal analyses are presented hereafter.

Reliable responsibility assignment (minimal cut-set) analysis. Policies are a tool for managers for creating a robust business process implementation system. In particular, lots of review and control responsibilities that are defined in organizations and described in our model through the use of operational assurance profiles, could be seen as redundant work. A traditional concept in reliability theory is the minimal cut-set of a system [30]. In our case, the cut-set of a process implementation is a set of goals that when no longer reliable can put at risk the business goals.

Our model allows 3 different minimal cut-sets analyses of the system. First, the minimal cut-sets of the process assurance profile that points out the different redundancies (such as a reviewing process). Second, the minimal cut-sets with the responsibility-for-achievement that represents a degree of redundancy: if the clerk fails to fulfil its responsibility, the manager might be able to mitigate that failure. Third, the minimal cut-sets in regard to the completeness of responsibilities concerning reliable commitments, accountabilities and required capabilities.

Separation-of-duties analyses [31] are made with minimal-cut sets. Indeed, separation-of-duties aims at increasing the minimal cardinality of all minimal-cut sets. So, the failure of the process implementation will require the failure of more independent resources (managers and clerks), hence decreasing its probability. This analysis is appropriate for non-intentional misbehaviours of managers and clerks. Basel II statistics indicates the majority of operational risks in financial institutions occur due to those weaknesses. Dishonest, fraudulent and criminal behaviours have a limited impact [32].

Minimal obligation set and least privilege analysis. The set of obligations can be analysed in order to be minimal in accordance to the business goals and in accordance to the required assurance profile. For each policy, the refinement of the

policy goal into responsibilities must be complete, and the set of responsibilities must be minimal (i.e. the policy goal is not entailed when removing one responsibility of the policy). Least privilege requires that the minimal set of rights be granted in order for the obligations being realizable. Moreover, the responsibility of accountability imposes that the usage of rights are accounted for. This implies that the minimal set of rights is also allowed. This actually depends on the selected assurance profile. For instance, in our case study, a clerk having the "responsibility-for-doing" of assessing the operational risk data will have the right of using time-slots for doing that just in case the selected assurance profile includes the resource allocation outcome of the process performance management purpose.

```
maybe_non_minimal_policy(Policy) :-
   bagof(GoalFormalDefinition #<=> B,
         Responsibility^(Goal^(Resource
         (rdf_db:rdf(Policy, has_responsibility, Responsibility),
          rdf_db:rdf(Responsibility, obligation_is_assigned_to, Resource),
          rdf_db:rdf(Responsibility, has_goal, Goal),
          rdf_db:rdf(Goal, has_formal_definition, GoalFormalDefinition)
         )), ConstraintSet),
   rdf_db:rdf(Policy, policy_has_goal, PolicyGoal),
   rdf_db:rdf(PoliyGoal, has_formal_definition, PolicyGoalFormalDefinition),
   length(ConstraintSet,L), L1 is L-1, \+ run_OTTER(
   select_constraint(ConstraintSet, L1), PolicyGoalFormalDefinition #<=> 0 ).
```

Fig. 7. SWI-Prolog code for identifying non-minimal policy obligation sets

The example shown in the Figure 7, detects a non-minimal obligation set by removing one obligation that do not produce any goal violation for the new refinement. (Note that out tool cannot to detect all non-minimal obligation sets due to the limitations of bounded model checking.) The Prolog predicate "select_constraint" just selects successively (on backtracking) all possible subsets of constraints having a cardinality given in the argument, and the predicate "run_OTTER" feeds the automated theorem prover with the formulae. (The namespaces of concepts are not shown for a better readability or the Prolog code.)

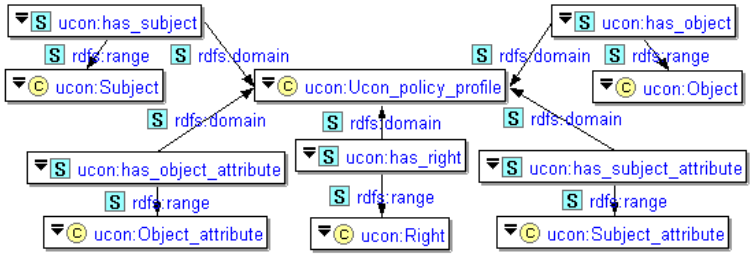

Fig. 8. Policy profile for usage control policies

Delegation of responsibilities analyses. [31] The delegation of responsibilities is implicitly handled in the model. Indeed, one can say that when a manager splits its "responsibility-for-achievement" into a number of "responsibility-for-doing" that are assigned to clerks, this is a kind of delegation. Dynamics aspects of the delegation can be defined in the temporal formulae. However, all delegation chains have their length and pattern fixed into the goal refinement model because, in

financial institutions, the length of delegation chains cannot be fully dynamic. Analyses on responsibility models (coded in Prolog) are described in [33], but no method is given for constructing those models from business processes and goals.

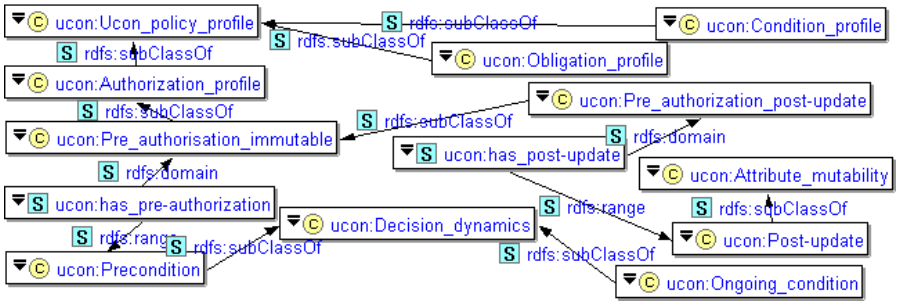

Fig. 9. Dynamics of usage right decisions and of attributes

4 The mapping onto policies for IT systems

It is not the intent of this section to describe in details this ontology, but it aims at showing the principles of the mapping between policies for managers and policies for IT systems. The mapping onto policies for IT systems is made through the Usage CONtrol policy family of models [34] because it can represents classical policy models and can be implemented, for instance, with XACML [35].

```
permission(Role, Object, Right) :-
    rdfs:rdfs_individual_of(Object,
                            'Work_product'),
    rdfs:rdfs_individual_of(Role, 'Resource'),
    rdf_db:rdf(Act, has_resource, Role),
    ((rdf_db:rdf(Act, has_input,Object),
      Right=input
     );(rdf_db:rdf(Act, has_output,Object),
        Right=output   )).
```

```
ucon_rbac_allowed(Subject, Object, Right) :-
    rdfs:rdfs_individual_of(Subject, 'Subject'),
    rdfs:rdfs_individual_of(Object, 'Object'),
    rdf_db:rdf(Practice, has_resource, Subject),
    ( rdf_db:rdf(Practice, has_input, Object) ;
      rdf_db:rdf(Practice, has_output, Object) ),
    rdf_db:rdf(Subject, has_role,Role),
    rdf_db:rdf(Object, has_work_product, Work_product),
    permission(Role, Work_product, Right).
```

Fig. 10. Prolog code querying the rights allowed to resources (subjects) for a target (object)

The policy model UCON (Figure 8) generalizes of the usual access control policy model to usage control policy model. This family of models (or meta-model) is based the well-known concepts: a *subject* gets some *rights* for *target resources*. However, the focus is no longer on how to structure the attributes of the subjects and resources (for instance, in a hierarchy of roles and permissions as in RBAC [34]), but to structure the decision concerning the rights. This is why in addition to *authorizations* there are *obligations* and external *conditions* that are defined. Moreover, the dynamics of the decision is considered through the concepts of *ongoing* decision controls, and the *mutability* of attributes. With the concept of policy profile, RBAC profile (not shown on the Figures) is a specialisation of the UCON profile “pre authorization immutable” (Figure 9). The Figure 10 illustrates the link between each ontology by giving the Prolog predicates that are used to query the rights allowed to resources (subjects) for a target (object). The RBAC concept of “role” is just mapped onto the concept “subject attribute” of Figure 8.

5 Conclusions and work plan

By using a set of concepts structuring the alignment of policies (goals, indicators, policies, strategies), structuring the assurance profile of business processes (purpose and outcomes), structuring the policies themselves, one can provide a set of analyses that help managers to build their policies, and allow experts to use lightweight goal-oriented formal analyses. The reliability of policy system can be formally defined and be the basis of usual requirements on policies such separation of duties, delegation of responsibilities, … To our knowledge, the operational assurance profile is not present in requirements engineering models of business processes. Although sometimes parts of the models concern operational assurance of the process, it is often too technical and/or spread within the model that makes difficult for managers to understand and analyse the models.

The work in progress concerns the definition of the value delivery assurance underlying good governance principles. The reliability of policy systems is an important basis for analysing the value delivery assurance. A tool is under construction, based on our ontology-based database. New real case studies in financial institutions are still in progress (e.g. a model of venture capital fund-of-funds management). The link between the our models and the technical policies is still under study with UCON in order to provide a technical layer for the policies defined by the managers which is based on new technologies, such as DRM.

6 References

[1] M. Hammer (1996) Beyond Reengineering: How the Process-Centered Organization is Changing Our Lives. HarperBusiness.
[2] A. Gunasekaran and B. Kobu (2002) Modelling and analysis of business process reengineering, Int. J. Prod. Res., 2002, vol. 40, no. 11, 2521:2546
[3] F. Baader et al., editors. (2003) The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press.
[4] IFRS: International Financial Reporting Standards, IASCF, USA. SoX: Sarbanes Oxley Act of 2002, USA. COSO: Internal Control – Integrated Framework, CSOTC.
[5] Basel Committee on Banking Supervision (2004) International Convergence of Capital Measurement and Capital Standards, Basel.
[6] R. Crook, D. Ince, B. Nuseibeh (2003) Modelling access policies using roles in requirements engineering, Information and Software Technology, 45:979-991.
[7] N. Damianou, A. Bandara, M. Sloman and E. Lupu, (2002) A survey of policy specification approaches, Imperial College of Science Technology
[8] and Medicine, London, (http://www.doc.ic. ac.uk/~mss/MSSPubs.html)
[9] A. Rifaut, M. Picard and B. Di Renzo (2006) ISO/IEC 15504 Process Improvement to Support Basel II Compliance of Operational Risk Management in Financial Institutions, International Conference SPiCE 2006
[10] CSSF (2006) http://www.cssf.lu/index.php?id=130
[11] A. Rifaut and C. Feltus (2006) Improving Operational Risk Management Systems by Formalizing the Basel II Regulation with Goal Models and the ISO/IEC 15504 Approach, REMO2V, CAISE06, Luxembourg.
[12] R.N. Anthony (1965) Planning and Control Systems: A Framework for Analysis. Harward University, Boston, USA.

[13] A. Rifaut (2005) Goal-Driven Requirements Engineering for Supporting the ISO 15504 Assessment Process, EuroSPI 2005, Budapest.
[14] J. Henderson and N. Venkatraman (1999) Strategic alignment: Leveraging technology for transforming organizations IBM Systems Journal : 38.
[15] Osterwalder and Pigneur (2005) An Ontology for e-business models. In Value Creation from E-Business Models, Wendy Currie ed., Butterworth-Heinenmann.
[16] W. Robson (1997) Strategic Management and Information Systems, Pitman.
[17] Chaffey et al. (2005) Business Information Systems: Technology, Development and Management for the E-business, Prentice Hall.
[18] E. Kavakli and P. Loucopoulos (2004) Goal Driven Requirements Engineering: Analysis and Critique of Current Methods, in Information Modeling Methods and Methodologies (Adv. topics of Database Research), 102:124
[19] Van Solingen (1999) The Goal/Question/Metric Method: A Practical Guide For Quality Improvement of Software Development McGraw-Hill,.
[20] P. Giorgini, N. Maiden, J. Mylopoulos, E. Yu (eds.) (2006) "Tropos/i*: Applications, variations and Extensions", Cooperative Information Systems Series, MIT Press.
[21] R. Kaplan and D. Norton (1996) The Balanced Scorecard. Harvard Bus. School Press
[22] B. Di Renzo, M. Hillairet, M. Picard, A. Rifaut, C. Bernard, D. Hagen, P. Maar, D. Reinard (2005) Operational Risk management in Financial Institutions: Process Assessment in Concordance with Basel II, International Conference SPiCE 2005.
[23] Rifaut A., (2005) An assessment method compliant to the Basel II regulation on operational risk management: example advocating that regulations can enhance innovation when based on quality goals, Proceedings of the conference New developments in Financial Planning Hochschule, Liechtenstein, December 2005.
[24] J. Dobson and J. McDermid.(1989) A Framework for Expressing Models of Security Policy. in IEEE Symposium on Security and Privacy. Oakland, CA.
[25] J. Dobson (1993) New Security Paradigms: What Other Concepts Do We Need as Well? In 1st New Security Paradigms Workshop. Little Compton: IEEE Press.
[26] C. Ponsard, P. Massonet, A. Rifaut, J.F. Molderez, A.l van Lamsweerde, H. Tran Van (2005) Early Verification and Validation of Mission Critical Systems. Electr. Notes Theor. Comput. Sci. 133: 237-254
[27] A. Rifaut, P. Massonet, J.F. Molderez, C. Ponsard, P. Stadnik, A. van Lamsweerde, H. Tran Van (2003) FAUST: Formal Analysis Using Specification Tools. RE 2003: 350
[28] SWI-Prolog-XPCE Semantic Web Library package (http://www.swi-prolog.org)
[29] T. Latvala, A. Biere, K. Heljanko, T.A.. Junttila, (2005) Simple Is Better: Efficient Bounded Model Checking for Past LTL. VMCAI 2005:380-395
[30] Kececioglu, D. (1991) Reliability Engineering Handbook, Vol. 2, Prentice Hall.
[31] A. Schaad and J. D. Moffett (2002) Delegation of Obligations, POLICY 2002.
[32] Basel Committee on Banking Supervision (2002) The 2002 Loss Data Collection Exercise for Operational Risk: Summary of the Data Collected. Basel.
[33] J. Moffett and M. Sloman (1993) Policy Hierarchies for Distributed Systems Management. IEEE Journal on Selected Areas in Communication, 11-9 : 1404–1414.
[34] J. Park and R. Sandhu (2004) The UCON-ABC Usage Control Model, ACM Transactions on Information and System Security, Vol. 7, No. 1 : 128–174.
[35] X. Zhang, M. Nakae, M.J. Covington, R. Sandhu (2005) A Usage-based Authorization Framework for Collaborative Computing Systems. ACM, SACMAT
[36] S. Hawke (2003) surnia -- OWL full reasoner based on otter, http://www.w3.org/2003/08/surnia/
[37] J. A. Kalman (2001) Automated Reasoning with Otter, Rinton Press.